

REMARKS

[0003] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-5, 10-15, 17, 20-23, 25, 27, and 30-34 are presently pending. Claims 1, 4, 11, 12, 13, 14, 15, 17, 21, 22, 23, and 25 are amended herein; claims 7, 24, and 27 are cancelled herein; and new claims 31-34 are added herein.

Statement of Substance of Interview

[0004] The Examiner graciously talked with me—the undersigned representative for the Applicant—on July 13, 2007. Applicant greatly appreciates the Examiner's willingness to talk. Such willingness is invaluable to both of us in our common goal of an expedited prosecution of this patent application.

[0005] During the interview, I discussed how the claims differed from the cited art, namely Legaria, Faiman, and Chengwen. I pointed out that none of the cited references disclose algebrizing a SQL tree by performing multiple operations during a single pass through the tree. I also pointed out that none of the cited references disclose performing constant folding as part of the algebrizing.

[0006] I understood the Examiner to suggest amending the claims to clarify what is meant by the term "operations", and that such clarification would help further prosecution of the case.

[0007] Applicant herein amends the claims in the manner discussed during the interview. Accordingly, Applicant submits that the pending claims are allowable over the cited art of record for at least the reasons discussed during the interview.

Formal Request for an Interview

[0008] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0009] Please contact me or my assistant to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for us, I welcome your call to either of us as well. Our contact information may be found on the last page of this response.

Claim Amendments and Additions

[0010] Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1, 4, 11, 12, 13, 14, 15, 17, 21, 22, 23, and 25 herein.

[0011] Furthermore, Applicant adds new claims 31-34 herein. These new claims are fully supported by the Application (and previously canceled claims) and therefore do not constitute new matter.

Formal Matters

Specification

[0012] Applicant has noted informalities in the specification. Accordingly, Applicant amends several paragraphs of the specification, as shown above, to correct these informalities.

Drawings

[0013] Applicant has noted informalities in Fig. 2. Specifically, Fig. 2 does not include reference number 202, which is given in the specification. Accordingly, Applicant submits herewith a replacement drawing to correct these informalities.

Substantive Matters

Claim Rejections under § 103

[0014] The Examiner rejects claims 1-5, 7, 10-15, 17, 20-25, 27, and 30 under §103. For the reasons set forth below, the Examiner has not shown made a *prima facie* case showing that the rejected claims are obvious.

[0015] Accordingly, Applicant respectfully requests that the §103 rejections be withdrawn and the case be passed along to issuance.

[0016] The Examiner's rejections are based upon the following references alone and/or in combination:

- **Legaria:** Legaria et al. Non-Patent Literature, "Orthogonal Optimization of Sub Queries and Aggregation";
- **Faiman:** U.S. Patent No. 5,836,014; and
- **Chengwen:** Chengwen et al. Non-Patent Literature, "A Framework for Global Optimization of Aggregated Queries."

Overview of the Application

[0017] The Application describes a technology for algebrizing a syntax tree representation of a relational database query into a relational algebra representation. The algebrizing is performed such that multiple processing steps are performed in a single pass through the syntax tree. Furthermore, the algebrizing may include a constant folding process.

Cited References

[0018] The Examiner cites Legaria as the primary reference, Faiman as the secondary reference, and Chengwen as the tertiary reference in the obviousness-based rejections.

Legaria

[0019] Legaria describes different types of query optimization.

Faiman

[0020] Faiman describes a constant-folding mechanism in a Multilanguage optimizing compiler.

Chengwen

[0021] Chengwen describes global optimization of aggregate queries.

Obviousness Rejections

Lack of *Prima Facie* Case of Obviousness (MPEP § 2142)

[0022] Applicant disagrees with the Examiner's obviousness rejections. Arguments presented herein point to various aspects of the record to demonstrate that all of the criteria set forth for making a *prima facie* case have not been met.

Based upon Legaria, Faiman, and Chengwen

[0023] The Examiner rejects claims 1-5, 7, 10-15, 17, 20-25, 27, and 30 under 35 U.S.C. § 103(a) as being unpatentable over Legaria in view of Faiman, and further in view of Chengwen. Applicant respectfully traverses the rejection of these claims and asks the Examiner to withdraw the rejection of these claims.

Independent Claim 1

[0024] As discussed during the telephone interview on July 13, 2007, Legaria does not disclose, "a method for algebrizing a syntax tree representation of a relational database query into a relational algebra representation," the method comprising "performing at least two operations in a single pass through said syntax tree representation," as recited in claim 1. Furthermore, the Examiner has not relied upon Faiman or Chengwen as teaching this element.

[0025] Legaria describes different types of query optimization. As illustrated in Figure 1, and described on page 272, Legaria describes performing the following:

- Algebrize into initial operator tree
- Remove correlations
- Simplify outerjoin
- Reorder GroupBy

Sections 1.3 and 2.1 of Legaria discuss algebrizing a SQL tree. However, Legaria does not appear to describe a particular method for algebrizing a SQL tree, but rather, focuses on optimization processing that is performed after the SQL tree has been algebrized. Legaria does not disclose, "a method for algebrizing a syntax tree representation of a relational database query into a relational algebra representation," the method comprising "performing at least two operations in a single pass through said syntax tree representation," as recited in claim 1. Accordingly, claim 1 is allowable over Legaria in view of Faiman and Chengwen, and Applicant respectfully requests that the 103 rejection be withdrawn.

Dependent Claims 2-5 and 31

[0026] These claims ultimately depend upon independent claim 1. As discussed above, claim 1 is allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Independent Claim 11

[0027] Claim 11 recites elements that are similar to those recited in claim 1. Accordingly, for reasons similar to those presented above with reference to claim 1, claim 11 is also allowable, and Applicant respectfully requests that the 103 rejection be withdrawn.

Dependent Claims 12-15, 17, 32, and 33

[0028] These claims ultimately depend upon independent claim 11. As discussed above, claim 11 is allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Independent Claim 10

[0029] None of the cited references disclose algebrizing a syntax tree representation of a relational database query into a relational algebra representation, wherein the algebrizing includes performing constant folding.

[0030] The Office Action indicates that Faiman is being relied upon as teaching constant folding. While Faiman may disclose constant folding, Faiman does not teach constant folding while algebrizing an SQL tree. As indicated in Fig.

4 of the Application, constant folding has typically been performed by a query processor, after and SQL tree is algebrized. As illustrated in Fig. 5 of the Application, the claims are directed to a scenario in which the constant folding is performed while algebrizing the SQL tree. This distinction does not appear to be disclosed by any of the cited references. Accordingly, claim 10 is allowable over Legaria in view of Faiman and further in view of Chengwen, and Applicant respectfully requests that the 103 rejection be withdrawn.

Independent Claims 20, 21, and 30

[0031] Claims 20, 21 and 30 recite elements that are similar to those recited in claim 10. Accordingly, for reasons similar to those presented above with reference to claim 10, claims 20, 21, and 30 are also allowable, and Applicant respectfully requests that the 103 rejection of these claims be withdrawn.

Dependent Claims 22, 23, 25, and 34

[0032] These claims ultimately depend upon independent claim 21. As discussed above, claim 21 is allowable. It is axiomatic that any dependent claim which depends from an allowable base claim is also allowable. Additionally, some or all of these claims may also be allowable for additional independent reasons.

Conclusion

[0033] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action.** Please call/email me or my assistant at your convenience.

Respectfully Submitted,

Dated: 7/23/07

By: Kayla D. Brant

Kayla D. Brant
Reg. No. 46576
(509) 324-9256 x242
kayla@leehayes.com
www.leehayes.com

My Assistant: Carly Bokarica
(509) 324-9256 x264
carly@leehayes.com